

## EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	2	717/124.ccls. and "software build" same modif\$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/13 11:58
L2	0	717/128.ccls. and "software build" same modif\$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/13 11:58
L3	1	717/162.ccls. and "software build" same modif\$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/13 11:58
L4	0	717/163.ccls. and "software build" same modif\$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/13 11:58
L5	1	717/140.ccls. and "software build" same modif\$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/13 11:59
L6	0	717/141.ccls. and "software build" same modif\$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/13 11:59
L7	1	717/142.ccls. and "software build" same modif\$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/13 11:59
L8	0	717/144.ccls. and "software build" same modif\$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/13 11:59
L9	0	717/143.ccls. and "software build" same modif\$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/13 11:59
L10	1	717/145.ccls. and "software build" same modif\$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/13 11:59
L11	0	717/146.ccls. and "software build" same modif\$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/13 12:00

## EAST Search History

L12	4	I1 I2 I3 I4 I5 I6 I7 I8 I9 I10 I11	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/13 12:00
S1	50	("20050044533" "5742754" "6292830" "5960196" "6898784" "5335342" "6416838" "5951697" "6243862" "5600789" "5655121" "5781720" "5878050" "5896494" "5991897" "6002868" "6028999" "6243835" "6874099" "7080356" "20020178281" "20030051186" "20030167422" "20030192009" "20030196191" "20040073662" "20040128652" "20040133880" "20050144593" "20060010426" "20060075302" "20060075303" "6028998" "6247126" "5363383" "5608894" "5717928" "6006022" "20040064268" "20040143830" "20060080638" "7047518" "20020091990" "20060059458" "6430741" "5572664" "5758062" "5877942" "5913023" "5938779" ). pn.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/13 11:57
S2	1	"5450416".pn.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/10 18:49
S3	32	S1 and build	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/10 19:04
S4	32	S1 and build and test	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/10 19:06
S5	19	S1 and build and test and compar\$4	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/10 19:06
S6	50	test\$3 with ((software program) adj1 build)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/11 12:53
S7	38	S6 and compar\$4	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2006/11/11 12:54



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

+test +software +build

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report](#)

Published since January 1995 and Published before August 2003

Terms used **test software build**Sort results by Display results 
☒ [Save results to a Binder](#)
☐ [Search Tips](#)
☐ [Open results in a new window](#)
Try an [/](#)

Try this

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

1 [Building testable software](#)

Lin Zucconi, Karl Reed

September 1996 **ACM SIGSOFT Software Engineering Notes**, Volume 21 Issue 5

Publisher: ACM Press

Full text available: [pdf\(591.39 KB\)](#)Additional Information: [full citation](#), [abstract](#), [index terms](#)

This paper examines a connection between well known specification, design, implementation method not to have been previously well-formulated. We refer to the fact that the use of finite state machine (specification, design, implementation and testing) is well known and documented. However, despite being more than twenty years old, there appears to be no detailed proposal for a consistent FSM-based ...

2 [How Microsoft builds software](#)

Michael A. Cusumano, Richard W. Selby

June 1997 **Communications of the ACM**, Volume 40 Issue 6

Publisher: ACM Press

Full text available: [pdf\(712.71 KB\)](#)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)3 [Computer security: Software security vulnerability testing in hostile environments](#)

Herbert H. Thompson, James A. Whittaker, Florence E. Mottay

March 2002 **Proceedings of the 2002 ACM symposium on Applied computing**

Publisher: ACM Press

Full text available: [pdf\(547.79 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Traditional Black box software testing can be effective at exposing some classes of software faults that tend to manifest readily using these techniques. The problem is that many security failures occur in the field, but are often neglected during testing because of the difficulty to simulate these conditions. It is not secure if it behaves securely under all operating environments. Hostile environments ...

**Keywords:** fault injection, software defect, software failure, software security, software testing

4 [Requirements elicitation for an intelligent software test environment for the physically challenged](#)

Warren Moseley

January 2000 **Proceedings of the 5th international conference on Intelligent user interfaces**

Publisher: ACM Press

Full text available: [pdf\(888.73 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper is about the elicitation of the requirements for an intelligent interface for a software test environment to accommodate the physically challenged (PC). This research explores the use of eye-tracking methods and interfaces that are especially enhanced for the PC. In addition these devices provide assistance in ...

Intelligent User Interface to such an environment. It was never a stated objective o ...

**Keywords:** Americans with Disabilities Act (ADA) of 1990, design patterns, digital manipulatives automation, knowledge acquisition, knowledge elicitation, object oriented architecture, physical software architecture

## 5 Specification-based testing of synchronous software



Ioannis Parisis, Farid Ouabdesselam

October 1996

**ACM SIGSOFT Software Engineering Notes , Proceedings of the 4th ACM SIGSOFT software engineering SIGSOFT '96**, Volume 21 Issue 6

**Publisher:** ACM Press

Full text available: [pdf\(912.85 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [cit](#)

Synchronous programming makes the implementation of reactive software easier and safer. Automodel-checking have been developed within the synchronous approach to prove the satisfaction these methods often require huge memory or time amounts. As a solution to that problem we propose techniques allowing for automatic test data generation. These techniques can be used independently

## 6 Daily build and feature development in large distributed projects



Even-André Karlsson, Lars-Göran Andersson, Per Leion

June 2000

**Proceedings of the 22nd international conference on Software engineering**

**Publisher:** ACM Press

Full text available: [pdf\(93.48 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [cit](#)

Daily build is a software development paradigm that originated in the PC industry to get control allowing the focus on end user requirements and code. The PC industry used daily build to avoid an environment without a strong development process. Ericsson Radio Systems has chosen to introduce end user requirements and code, but from a different starting point with a traditional ...

**Keywords:** daily build, extreme programming, feature teams, incremental development, integration

## 7 Software testing: a machine learning experiment



Thomas J. Cheatham, Jungsoon P. Yoo, Nancy J. Wahl

February 1995

**Proceedings of the 1995 ACM 23rd annual conference on Computer science**

**Publisher:** ACM Press

Full text available: [pdf\(655.39 KB\)](#)

Additional Information: [full citation](#), [references](#), [index terms](#)

## 8 Regression test selection for Java software



Mary Jean Harrold, James A. Jones, Tongyu Li, Donglin Liang, Alessandro Orso, Maikel Pennings, Sridhar Gujarathi

October 2001

**ACM SIGPLAN Notices , Proceedings of the 16th ACM SIGPLAN conference on systems, languages, and applications OOPSLA '01**, Volume 36 Issue 11

**Publisher:** ACM Press

Full text available: [pdf\(292.35 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [cit](#)

Regression testing is applied to modified software to provide confidence that the changed parts have not been adversely affected by the modifications. To reduce the cost of regression test suite that was used to test the original version of the software---this process is called regression test selection algorithm selects every test case in the test suite that may reveal ...

## 9 Book Reviews: The Craft of Software Testing: Subsystem Testing Including Object-Based :



Maxick

John A. Kostecki


May 1998



**ACM SIGSOFT Software Engineering Notes**, Volume 23 Issue 3

**Publisher:** ACM Press

Full text available:

Additional Information:



 [pdf\(213.51 KB\)](#)[full citation](#)**10** The chaining approach for software test data generation

 Roger Ferguson, Bogdan Korel  
 January 1996 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volu  
**Publisher:** ACM Press  
 Full text available:  [pdf\(1.53 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cit](#)

Software testing is very labor intensive and expensive and accounts for a significant portion of s  
 testing process could be automated, the cost of developing software could be significantly reduc  
 is the process of identifying a set of test data that satisfies a selected testing criterion, such as s  
 this article we present a chaining approach for automat ...

**Keywords:** data dependency, dynamic analysis, heuristics, program execution

**11** Two case studies of open source software development: Apache and Mozilla

 July 2002 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volu  
**Publisher:** ACM Press  
 Full text available:  [pdf\(373.10 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cit](#)

According to its proponents, open source style software development has the capacity to compe  
 displace, traditional commercial development methods. In order to begin investigating such clai  
 source projects, the Apache web server and the Mozilla browser. By using email archives of sou  
 we quantify aspects of developer participation, core team size, code ownership, productivit ...

**Keywords:** Apache, Mozilla, Open source software, code ownership, defect density, repair inter


**12** Validation and verification: Deriving models of software fault-proneness

 Giovanni Denaro, Sandro Morasca, Mauro Pezzè  
 July 2002 **Proceedings of the 14th international conference on Software engineering**  
**Publisher:** ACM Press  
 Full text available:  [pdf\(164.38 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cit](#)

The effectiveness of the software testing process is a key issue for meeting the increasing dema  
 costs of software development. The estimation of software fault-proneness is important for asse  
 planning and tuning the testing process. Unfortunately, no general techniques are available for  
 distribution of faults to identify the correct level of test for the required qualit ...

**Keywords:** cross-validation, fault-proneness models, logistic regression, software faultiness, sc



**13** A perspective on teaching software testing

Edward L. Jones, Christy L. Chatmon  
 March 2001 **Journal of Computing Sciences in Colleges**, Volume 16 Issue 3  
**Publisher:** Consortium for Computing Sciences in Colleges , Consortium for Computing Sciences in Colleges  
 Full text available:  [pdf\(74.00 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [in](#)

This paper addresses the issue of how to approach the teaching of software testing. Relative to  
 software testing receives very little attention in the undergraduate curriculum. It is not practical  
 testing, so relevant test experiences need to be given throughout core courses. This paper pres  
 five essential principles that motivate the types of testing experiences a stud ...

**14** Experience reports: testing and fault correction: Assessing test-driven development at IBM

E. Michael Maximilien, Laurie Williams  
 May 2003 **Proceedings of the 25th International Conference on Software Engineering**  
**Publisher:** IEEE Computer Society

Full text available:  [pdf\(572.29 KB\)](#)  [Publisher Site](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [cit](#)

In a software development group of IBM Retail Store Solutions, we built a non-trivial software specification using a disciplined, rigorous unit testing and build approach based on the test-drive practice, we reduced our defect rate by about 50 percent compared to a similar system that was. The project completed on time with minimal development productivity impact. Additionally, ...



# 15 Technical papers: testing II: A framework for component deployment testing

Antonia Bertolino, Andrea Polini

May 2003

**Proceedings of the 25th International Conference on Software Engineering**

**Publisher:** IEEE Computer Society

Full text available:  [pdf\(1.34 MB\)](#)  [Publisher Site](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [inc](#)

Component-based development is the emerging paradigm in software production, though severe. In particular, the "component trust problem" refers to how adequate guarantees and documents transferred from the component developer to its potential users. The capability to test a component application environment can help establish the compliance of a candidate component to the customer.

# 16 Session: Extreme embedded a report from the front line




Gary Mueller, Janet Borzuchowski

November 2002

**OOPSLA 2002 Practitioners Reports**

**Publisher:** ACM Press

Full text available:  [pdf\(145.69 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [index terms](#)

Many embedded development environments are stuck somewhere in the backwash of software development practices, with the artifacts of those methodologies liberally scattered about, are the norm in the industry since structured methodologies first emerged, with shorter and shorter market windows and increasing more and more capabilities at lower and lower cost. Embedded system development must also adapt.

**Keywords:** eXtreme programming, embedded

# 17 Software metrics: roadmap




Norman E. Fenton, Martin Neil

May 2000

**Proceedings of the Conference on The Future of Software Engineering**

**Publisher:** ACM Press

Full text available:  [pdf\(1.25 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

**Keywords:** Bayesian belief nets, casual models, multi-criteria decision aid, risk assessment, software

# 18 A safe, efficient regression test selection technique




Gregg Rothermel, Mary Jean Harrold

April 1997

**ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 6, Number 2, April 1997

**Publisher:** ACM Press

Full text available:  [pdf\(730.74 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [cit](#)

Regression testing is an expensive but necessary maintenance activity performed on modified software. If the tests are correct and do not adversely affect other portions of the software. A regression test selection technique that is deemed necessary to validate modified software. We present a new technique for constructing control flow graphs for a procedure or program and its modified version ...

**Keywords:** regression test selection, regression testing, selective retest

# 19 Improving validation activities in a global software development



Christof Ebert, Casimiro Hernandez Parro, Roland Suttels, Harald Kolarczyk

July 2001

**Proceedings of the 23rd International Conference on Software Engineering**

**Publisher:** IEEE Computer Society

Full text available:

 [pdf\(288.74 KB\)](#)  [Publisher Site](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [in](#)

*Global software development challenges traditional techniques of software engineering, such as teamwork and coaching of engineers highly contribute towards successful projects. We will evaluate validation activities in a global setting within Alcatel's Switching and Routing business. We will introduce collocated inspections, intensive coaching, and feature-oriented development ...*


**Keywords:** *coaching, cost of non-quality, defect detection, efficiency, feature development, global inspection, teamwork, validation*

## 20 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on**

**Publisher:** IBM Press

Full text available:  [pdf\(4.21 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [in](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on process a better understanding of the execution of the application. The visualization tool we use is Poet, of Waterloo. However, these diagrams are often very complex and do not provide the user with our experience, such tools display repeated occurrences of non-trivial commun ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)